I. Amendments to the Claims

This listing of claims replaces without prejudice all prior versions and listings of claims in the application:

Listing of Claims:

1. (Currently Amended) A ~~connection management~~ system for managing a telecommunications connection comprising:

a first user terminal;

a second user terminal;

a telecommunications network interconnecting said first user terminal and said second user terminal; and

modular connection management software including a connection management software proxy for each of (i) said first user terminal, (ii) said second user terminal, and (iii) said telecommunications network, each of said connection management software proxies including at least one user reactor object and at least one connection agent and being operable to establish a connection between controlling applications executing on said user terminals,

~~to execute on said system; and~~

~~to provide the functionality to represent its owner's interests in managing the telecommunications connection.~~

whereby user reactor objects react to events and

where required instantiate a connection object and evolve into a connection agent which becomes responsible for the connection for the duration of the connection.

2. (Currently Amended) A system as claimed in ~~claim 1~~ claim 14, wherein each said proxy is operable to provide the functionality to represent its owner's interests in managing set up, maintenance and tear down of the telecommunications connection.

3. (Original) A system as claimed in claim 2, wherein each said proxy is operable to provide the functionality to represent its owner's interests with respect to cost and quality of service in managing set up, maintenance and tear down of the telecommunications connection.

Claim 4 (Cancelled)

5. (Currently Amended) A system as claimed in ~~claim 4~~ claim 3, wherein said telecommunications network comprises multiple telecommunications networks with varied protocols, each of said multiple telecommunications networks having its own connection management software proxy.

6. (Original) A system as claimed in claim 5,

wherein said proxy object is operable to locate said multiple software agents for execution on devices closest to where they are required.

7. (Original) A system as claimed in claim 6, which is further capable of handling multiple communications for a given entity, comprising:

a multitasking operating system executing on said system; and

said connection management software proxy for said given entity being operable to instantiate software agents require for each of said multiple communications.

8. (Original) A system as claimed in claim 7, wherein each said operating system comprises:

a real-time multitasking operating system including a scheduler to administer timely execution of software threads and functions.

9. (Original) A system as claimed in claim 8, wherein each said operating system comprises an operating system having a standard applications programming interface (API).

10. (Currently Amended) A method of ~~connection management for~~ managing a telecommunications system,

executing on a telecommunications server, comprising the steps of:

interconnecting a first user terminal and a second user terminal; and

executing modular connection management software including a connection management software proxy for each of (i) said first user terminal, (ii) said second user terminal, and (iii) said telecommunications server, each of said connection management software proxies including at least one user reactor object and at least one connection agent and being operable to establish a connection between controlling applications executing on said user terminals~~:~~

~~to execute on said telecommunications system; and~~

~~to provide the functionality to represent its owner's interests in managing a telecommunications connection~~,

whereby user reactor objects react to events and where required instantiate a connection object and evolve into a connection agent which becomes responsible for the connection for the duration of the connection.

11. (Currently Amended) A telecommunications server for a telecommunications system comprising:

interconnecting means for interconnecting a first user terminal and a second user terminal; and

modular connection management means including a connection management software proxy for each of (i) said

first user terminal, (ii) said second user terminal, and (iii) said telecommunications server, each of said connection management software proxies including at least one user reactor object and at least one connection agent and being operable to establish a connection between controlling applications executing on said user terminals, ~~:~~

~~to execute on said telecommunications system; and~~

~~to provide the functionality to represent its owner's interests in managing a telecommunications connection.~~

whereby user reactor objects react to events and where required instantiate a connection object and evolve into a connection agent which becomes responsible for the connection for the duration of the connection.

12. (Currently Amended) A ~~computer data signal embodied in a~~ carrier wave~~, said computer data signal comprising a set of machine executable~~ embodying computer readable code ~~being executable by~~ which when loaded enables a computer to perform the steps of ~~claim 10~~ claim 15.

13. (Currently Amended) A computer readable storage medium storing a set of machine executable code, said set of machine executable code being executable by a computer server to perform the steps of ~~claim 10~~ claim 15.

14. (New) A system as claimed in claim 11, wherein:

a controlling application on one user terminal contacts its proxy by opening a channel, which channel locates the proxy using a unique identifier to identify a service broker with which the proxy is associated;

the located proxy creates a user reactor and passes it the channel;

the controlling application sends a message to the user reactor via the channel, identifying the other participants in the connection;

the user reactor then causes the service broker to create a new connection object that includes a parameter which is a reference to the user proxy and the proxies for the other participants to the connection object;

the connection object contacts the user proxy of

each of the connection's participants and requests a connection agent for each;

each proxy evolves its user reactor into a connection agent;

the connection agents and connection object exchange events and messages;

the connection object creates a floor object which takes a list of the participant's connection agents and handles a negotiation; and

the connection agents then send the result of the negotiation to their corresponding controlling applications.

15. (New) A method as claimed in claim 10, wherein:

a controlling application on one user terminal contacts its proxy by opening a channel, which channel locates the proxy using a unique identifier to identify a service broker with which the proxy is associated;

the located proxy creates a user reactor and passes it the channel;

the controlling application sends a message to the user reactor via the channel, identifying the other participants in the connection;

the user reactor then causes the service broker to create a new connection object that includes a parameter which is a reference to the user proxy and the proxies for the other participants to the connection object;

the connection object contacts the user proxy of each of the connection's participants and requests a connection agent for each;

each proxy evolves its user reactor into a connection agent;

the connection agents and connection object exchange events and messages;

the connection object creates a floor object which takes a list of the participant's connection agents and handles a negotiation; and

the connection agents then send the result of the negotiation to their corresponding controlling applications.

16. (New) A telecommunications server as claimed in claim 10, wherein:

a controlling application on one user terminal contacts its proxy by opening a channel, which channel locates the proxy using a unique identifier to identify a service broker with which the proxy is associated;

the located proxy creates a user reactor and passes it the channel;

the controlling application sends a message to the user reactor via the channel, identifying the other participants in the connection;

the user reactor then causes the service broker to create a new connection object that includes a parameter which is a reference to the user proxy and the proxies for the other participants to the connection object;

the connection object contacts the user proxy of each of the connection's participants and requests a connection agent for each;

each proxy evolves its user reactor into a connection agent;

the connection agents and connection object exchange events and messages;

the connection object creates a floor object which takes a list of the participant's connection agents and handles a negotiation; and

the connection agents then send the result of the negotiation to their corresponding controlling applications.